

Especificaciones de Entrada y Salida

1. Aspectos generales de los programas y envio

- Los lenguajes de programacion aceptados para realizar la tarea son C , C++ o Java.
- El ejecutable se debe llamar “tarea1” para C o C++. La clase en Java que posee la funcion main debe llamarse “Tarea1”
- Esta permitido usar cualquier libreria externa, SOLO para parsear el patron prosite. Para cualquier otra actividad esta prohibido su uso, y va a ser sancionado. La libreria tiene que ser subida en Material Alumnos(si no ha sido ya subida), y tienen que incluir una breve explicacion de como usar la libreria en la entrega de la tarea y en la libreria subida.
- Incluir un Makefile para compilar el codigo en C o C++. Para Java, debe existir el archivo tarea1.java

2. Entrada de la tarea

El programa debe recibir primero el patron prosite, y luego debe pedir proteinas hasta que reciba “#”. La entrada debe ser leida en la entrada estandar.

3. Salida de la tarea

La salida del programa debe ser la salida estandar

3.1. Aspectos generales

- La salida del programa puede tener tantos espacios en blanco como quieran, ya que no seran tomados en cuenta, salvo los saltos de linea. O sea:

```
{a      ,   a}  
{a,a}
```

son la “misma” linea. Tambien no sera considerado las lineas en blanco.

- El orden de la salida es: Expresion regular, AFND, AFD y Resultados matching proteinas

3.2. Expresiones regulares

- La sintaxis de esta seccion es:

ER = *Expresion Regular* \n

- La sintaxis de la expresion regular es:

Simbolo	Significado
.	Concatenacion (Opcional ponerlo)
	Union
*	Clausura de Kleen
()	Parentesis
A	Letra del alfabeto
x	cualquier letra del alfabeto
e	epsilon
0(cero)	vacio

Es importante señalar que las letras mayusculas representan letras del alfabeto, mientras que las minusculas no representan nada, salvo la e y x.

- Un fin de linea representa el fin de la expresion regular, y por ende el de esta seccion

3.3. AFND y AFD

- La sintaxis de esta seccion es(cambiado AFND a AFD cuando se este mostrando el Automata Finito Deterministico):

AFND= \n
K= *Conjunto de estados* \n
s= *Estado inicial* \n
F= *Conjunto de estados finales* \n
Lista de transiciones

- El Conjunto K se puede representar de dos formas:

- Normal:

{estado1, estado23, estado89}

Los nombres de estados solo pueden tener caracteres alfanumericos

- Abreviada:

{2...10}

Que significa que existe el estado 2, 3,..., hasta 10. Los nombres de los estados en esta forma estan restringidos solo a numeros

- La sintaxis de una linea de la lista de transiciones es:

$D(\text{estado}, \text{letra}) = \text{Estado}(s) \text{ de transicion}$

En un AFD, la letra es una del alfabeto y el estado de transicion es un estado (y no un conjunto de estados)

En un AFND, la letra es una del alfabeto o epsilon(e) y los estados de transicion son un conjunto de estados

3.4. Resultado matching de proteinas

- La sintaxis de esta seccion es:

Proteina m : *Conjunto de posiciones*

4. Ejemplos de Entrada y Salida

4.1. Ejemplo de entrada

```
A-C-C(2,3)-[AG]-x(0,1)-G\n
ACGTACCTTTAGG\n
GTTACCTAGGC\n
#\n
```

4.2. Ejemplo de salida

La entrada es:

```
A(0,1)-x-G\n
TTTTTTTTTTT\n
ATG\n
TTTAGGTT\n
#\n
```

La salida es:

```
ER = (A|e) x G \n
AFND=\n
K = {euforia, alegria, tristesa, pena} \n
s= euforia \n
F= {tristesa} \n
D(euforia, A)={alegria}\n
D(euforia, e)={alegria}\n
D(alegria, A)={ pena }\n
D(alegria, C)={ pena }\n
D(alegria, D)={ pena }\n
D(alegria, E)={ pena }\n
D(alegria, F)={ pena }\n
D(alegria, G)={ pena }\n
```

```
D(alegria, H)={ pena }\n
D(alegria, I)={ pena }\n
D(alegria, K)={ pena }\n
D(alegria, L)={ pena }\n
D(alegria, M)={ pena }\n
D(alegria, N)={ pena }\n
D(alegria, O)={ pena }\n
D(alegria, P)={ pena }\n
D(alegria, Q)={ pena }\n
D(alegria, R)={ pena }\n
D(alegria, S)={ pena }\n
D(alegria, T)={ pena }\n
D(alegria, U)={ pena }\n
D(alegria, V)={ pena }\n
D(alegria, W)={ pena }\n
D(alegria, Y)={ pena }\n
D(pena,G)={tristes}\n
\n
\n
AFD=\n
  K={1...6}\n
  s=1  \n
  F={4,5}\n
  D(1, A) =2 \n
  D(1, C)=3  \n
D(1, D)=3  \n
D(1, E)=3  \n
D(1, F)=3  \n
D(1, G)=3  \n
D(1, H)=3  \n
D(1, I)=3  \n
D(1, K)=3  \n
D(1, L)=3  \n
D(1, M)=3  \n
D(1, N)=3  \n
D(1, O)=3  \n
D(1, P)=3  \n
D(1, Q)=3  \n
D(1, R)=3  \n
D(1, S)=3  \n
D(1, T)=3  \n
D(1, U)=3  \n
D(1, V)=3  \n
D(1, W)=3  \n
D(1, Y)=3  \n
D(2, A) =3  \n
```

D(2, C)=3 \n
D(2, D)=3 \n
D(2, E)=3 \n
D(2, F)=3 \n
D(2, G)=4 \n
D(2, H)=3 \n
D(2, I)=3 \n
D(2, K)=3 \n
D(2, L)=3 \n
D(2, M)=3 \n
D(2, N)=3 \n
D(2, O)=3 \n
D(2, P)=3 \n
D(2, Q)=3 \n
D(2, R)=3 \n
D(2, S)=3 \n
D(2, T)=3 \n
D(2, U)=3 \n
D(2, V)=3 \n
D(2, W)=3 \n
D(2, Y)=3 \n
D(3, A) =6 \n
D(3, C)=6 \n
D(3, D)=6 \n
D(3, E)=6 \n
D(3, F)=6 \n
D(3, G)=5 \n
D(3, H)=6 \n
D(3, I)=6 \n
D(3, K)=6 \n
D(3, L)=6 \n
D(3, M)=6 \n
D(3, N)=6 \n
D(3, O)=6 \n
D(3, P)=6 \n
D(3, Q)=6 \n
D(3, R)=6 \n
D(3, S)=6 \n
D(3, T)=6 \n
D(3, U)=6 \n
D(3, V)=6 \n
D(3, W)=6 \n
D(3, Y)=6 \n
D(4, A)=6 \n
D(4, C)=6 \n
D(4, D)=6 \n

D(4, E)=6 \n
D(4, F)=6 \n
D(4, G)=5 \n
D(4, H)=6 \n
D(4, I)=6 \n
D(4, K)=6 \n
D(4, L)=6 \n
D(4, M)=6 \n
D(4, N)=6 \n
D(4, O)=6 \n
D(4, P)=6 \n
D(4, Q)=6 \n
D(4, R)=6 \n
D(4, S)=6 \n
D(4, T)=6 \n
D(4, U)=6 \n
D(4, V)=6 \n
D(4, W)=6 \n
D(4, Y)=6 \n
D(5, A)=6 \n
D(5, C)=6 \n
D(5, D)=6 \n
D(5, E)=6 \n
D(5, F)=6 \n
D(5, G)=6 \n
D(5, H)=6 \n
D(5, I)=6 \n
D(5, K)=6 \n
D(5, L)=6 \n
D(5, M)=6 \n
D(5, N)=6 \n
D(5, O)=6 \n
D(5, P)=6 \n
D(5, Q)=6 \n
D(5, R)=6 \n
D(5, S)=6 \n
D(5, T)=6 \n
D(5, U)=6 \n
D(5, V)=6 \n
D(5, W)=6 \n
D(5, Y)=6 \n
D(6, A)=6 \n
D(6, C)=6 \n
D(6, D)=6 \n
D(6, E)=6 \n
D(6, F)=6 \n

```
D(6, G)=6  \n
D(6, H)=6  \n
D(6, I)=6  \n
D(6, K)=6  \n
D(6, L)=6  \n
D(6, M)=6  \n
D(6, N)=6  \n
D(6, O)=6  \n
D(6, P)=6  \n
D(6, Q)=6  \n
D(6, R)=6  \n
D(6, S)=6  \n
D(6, T)=6  \n
D(6, U)=6  \n
D(6, V)=6  \n
D(6, W)=6  \n
D(6, Y)=6  \n
Proteina 1: {}  \n
Proteina 2: {3} \n
Proteina 3: {5,6} \N
```